

FIG. 1

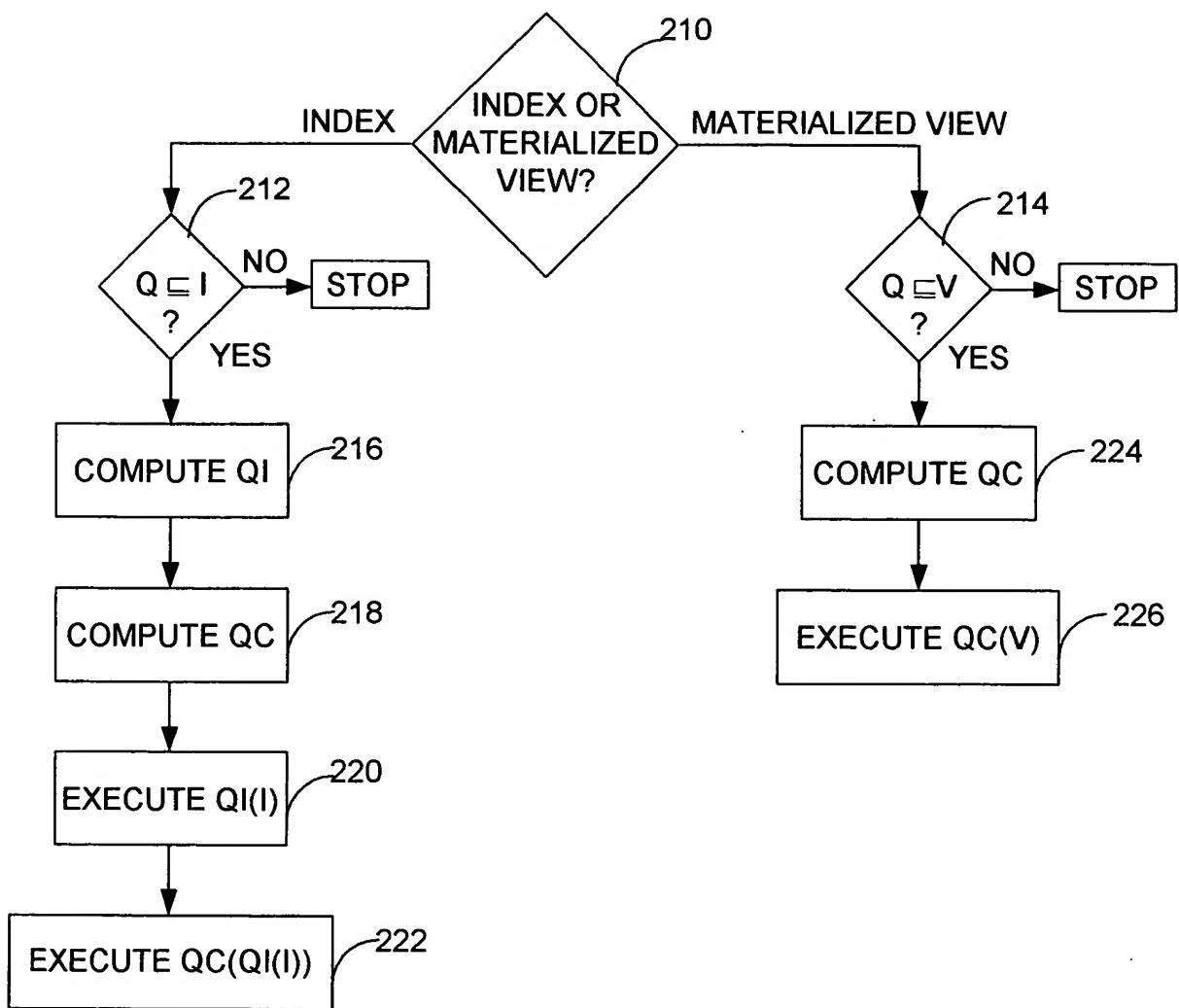


FIG. 2

P = //*[@*]

**Q₁ = //order/lineitem
[@price and discount]**

**Q₂ = //lineitem
[@price or price]**

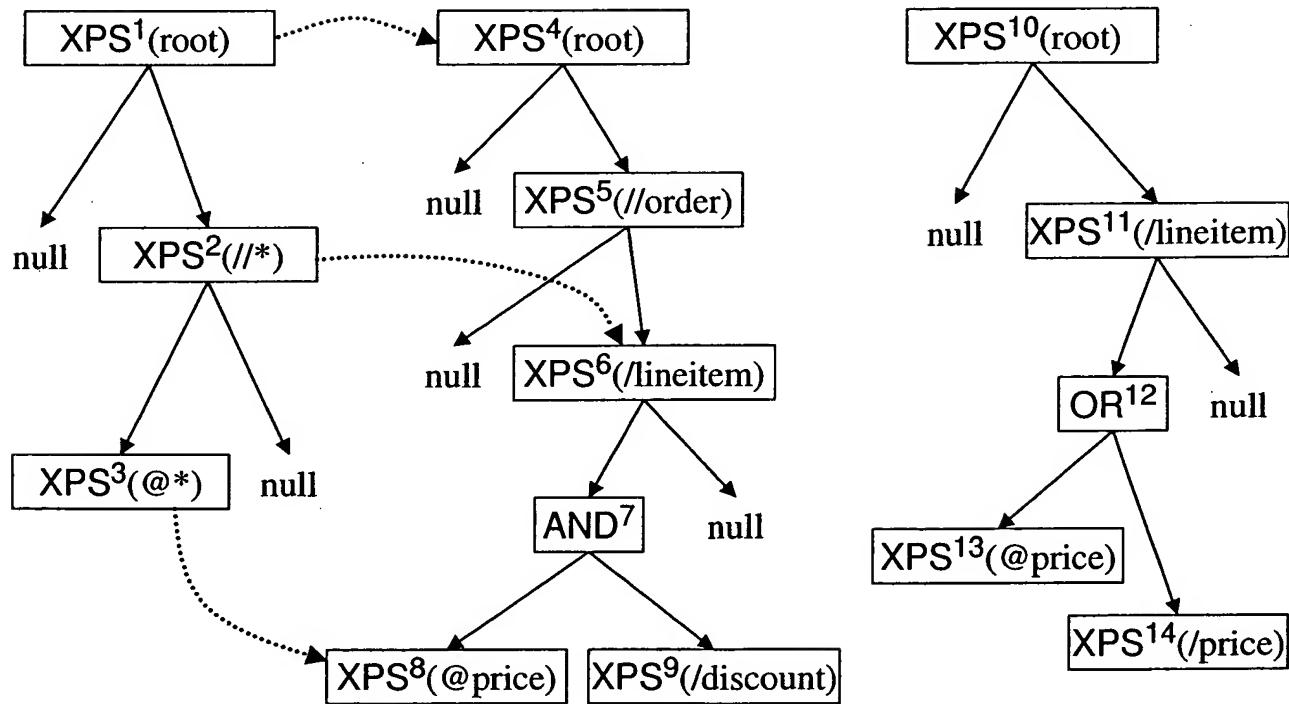


FIG. 3

1	$\text{matchStep}(p, q)$	
1.1	if ($q = q_1 \wedge q_2$)	$\text{matchStep}(p, q) \rightarrow \text{matchStep}(p, q_1) \vee \text{matchStep}(p, q_2)$
1.2	else if ($q = q_1 \vee q_2$)	$\text{matchStep}(p, q) \rightarrow \text{matchStep}(p, q_1) \wedge \text{matchStep}(p, q_2)$
1.3	else if ($p_{\text{axis}} = \text{"descendant"}$)	$\text{matchStep}(p, q) \rightarrow \bigvee \{\text{matchChildren}(p, c)\}, \forall c \in \{\text{preorder traversal of } q\},$ such that $c_{\text{axis}} \neq \text{"attribute"}$
1.4	else if ($p_{\text{axis}} = q_{\text{axis}}$)	$\text{matchStep}(p, q) \rightarrow \text{matchChildren}(p, q)$
1.5	else	$\text{matchStep}(p, q) \rightarrow \text{False}$
2	$\text{matchChildren}(p, q)$	
2.1	if ($p_{\text{test}} = \text{"*"} \vee (p_{\text{test}} = q_{\text{test}})$)	$\text{matchChildren}(p, q) \rightarrow \text{matchPred}(p_{\text{pred}}, q) \wedge \text{matchNext}(p_{\text{next}}, q)$
2.2	else	$\text{matchChildren}(p, q) \rightarrow \text{False}$
3	$\text{matchPred}(p_{\text{pred}}, q)$	
3.1	if ($p_{\text{pred}} = \text{null}$)	$\text{matchPred}(p_{\text{pred}}, q) \rightarrow \text{True}$
3.2	else if ($q = \text{null}$)	$\text{matchPred}(p_{\text{pred}}, q) \rightarrow \text{False}$
3.3	else if ($p_{\text{pred}} = p_1 \wedge p_2$)	$\text{matchPred}(p_{\text{pred}}, q) \rightarrow \text{matchPred}(p_1, q) \wedge \text{matchPred}(p_2, q)$
3.4	else if ($p_{\text{pred}} = p_1 \vee p_2$)	$\text{matchPred}(p_{\text{pred}}, q) \rightarrow \text{matchPred}(p_1, q) \vee \text{matchPred}(p_2, q)$
3.5	else	$\text{matchPred}(p_{\text{pred}}, q) \rightarrow \text{matchStep}(p_{\text{pred}}, q_{\text{next}})$
4	$\text{matchNext}(p_{\text{next}}, q)$	
4.1	if ($p_{\text{next}} = \text{null}$)	$\text{matchNext}(p_{\text{next}}, q) \rightarrow \text{True}$
4.2	else if ($q = \text{null}$)	$\text{matchNext}(p_{\text{next}}, q) \rightarrow \text{False}$
4.3	else	$\text{matchNext}(p_{\text{next}}, q) \rightarrow \text{matchStep}(p_{\text{next}}, q_{\text{pred}}) \vee \text{matchStep}(p_{\text{pred}}, q_{\text{next}})$

FIG. 4

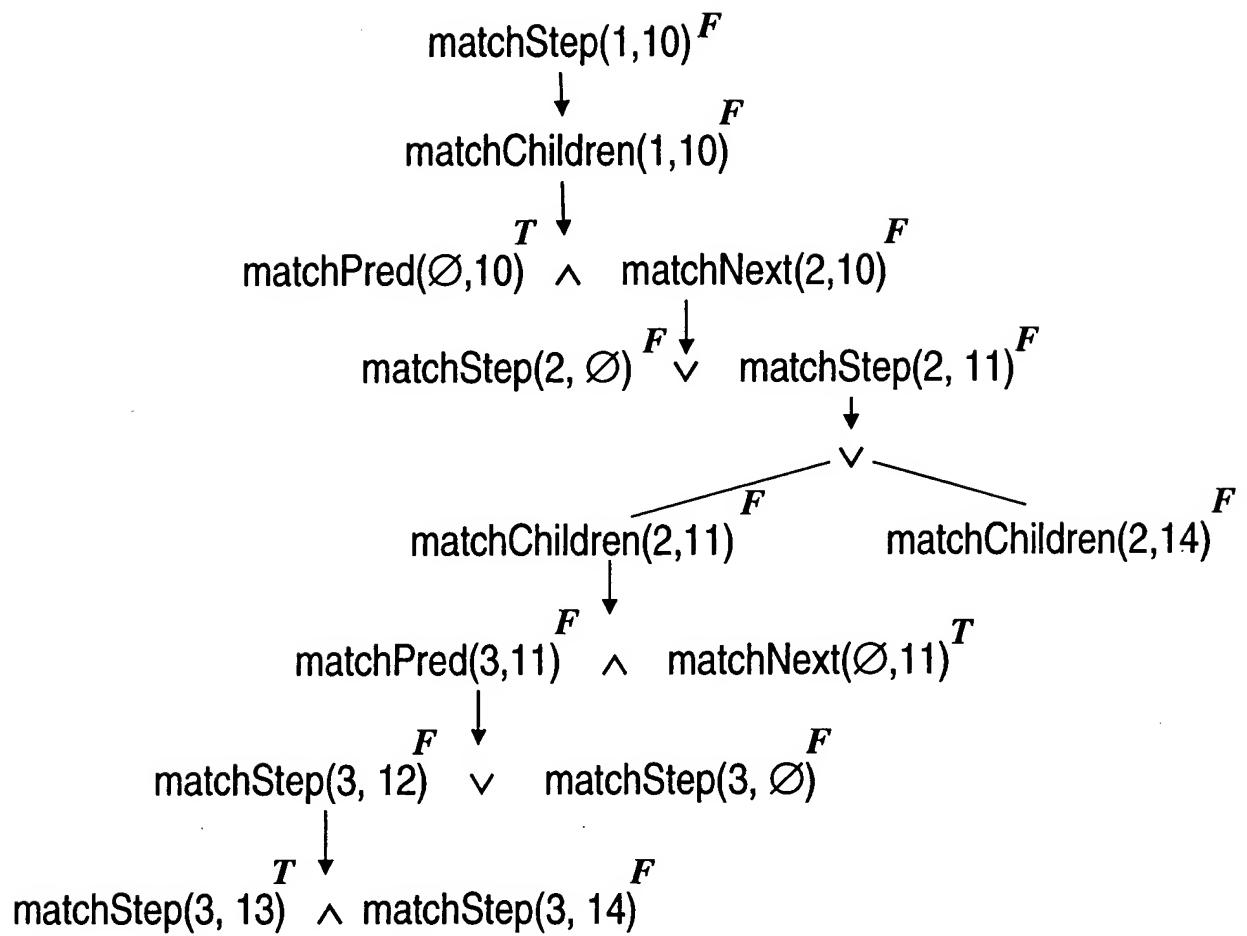
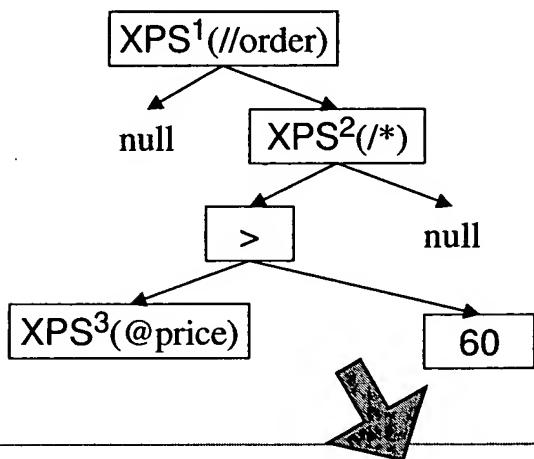
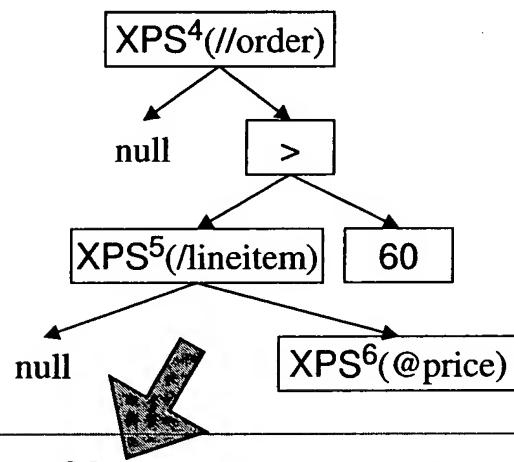


FIG. 5

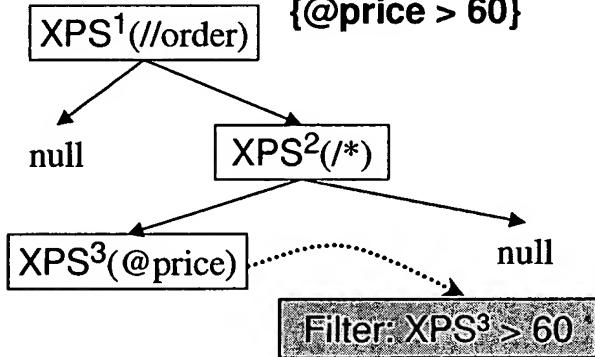
$P = //\text{order}/*[@\text{price} > 60]$



$Q = //\text{order}[\text{lineitem}/@\text{price}>100]$



$P' = //\text{order}/*[@\text{price}],
{@\text{price} > 60}$



$Q' = //\text{order}[\text{lineitem}/@\text{price}],
{@\text{price} > 100}$

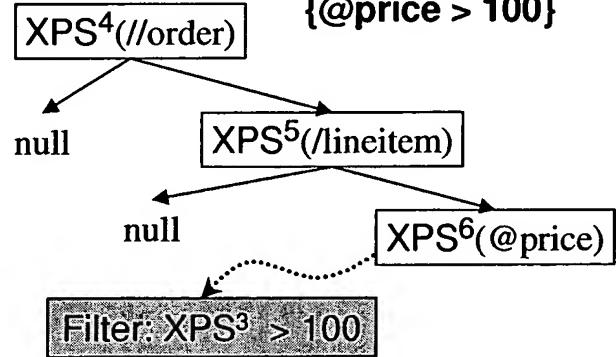


FIG. 6

ExtractPredicates(E)

```
foreach c in breadth-first traversal of E do
    if (c ∈ <, <, >, >, =, ≠) then
        // left extraction point
        lep = c.left;
    if (lep is an XPS) // traverse next's
        while (lep.next ≠ null) lep=lep.next;
        // right extraction point
        rep=c.right;
    if (rep is an PS) // traverse next's
        while (rep.next ≠ null) rep=rep.next;
        add "c (lep, rep)" to the filter list
    if (lep is a const) ∧ (rep is an XPS)
        replace c with the right child
    if (rep is a const) ∧ (lep is an XPS)
        replace c with left child
    if (lep is an XPS) ∧ (rep is an XPS)
        replace comparison in c with AND
```

FIG. 7

$P = //\text{employee}[\cdot//\text{employee}/@\text{salary} < \cdot//\text{employee}/@\text{bonus}]//\text{employee}$
 $Q = //\text{employee}[\text{employee}/@\text{salary} < \text{employee}[\text{employee}/@\text{salary} < \text{employee}/@\text{bonus}]/@\text{bonus}]//\text{employee}$

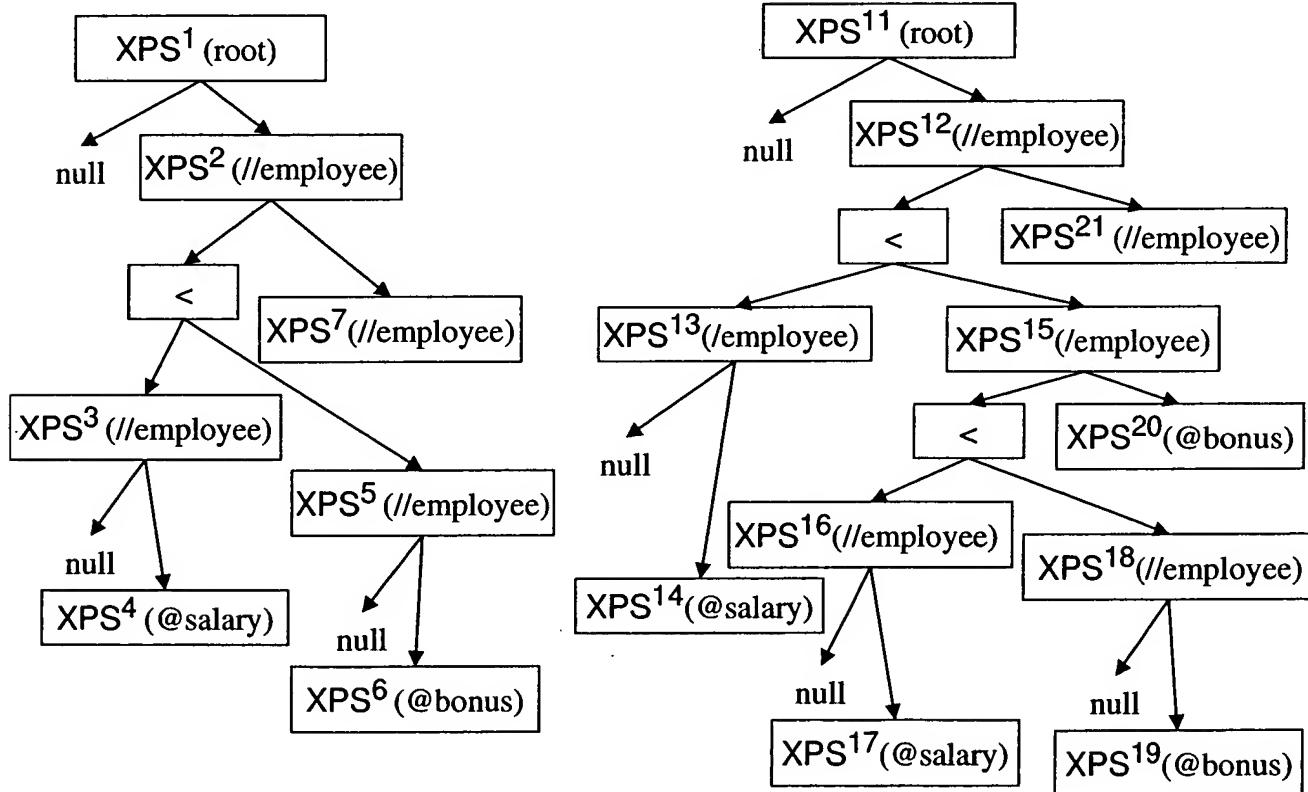


FIG. 8

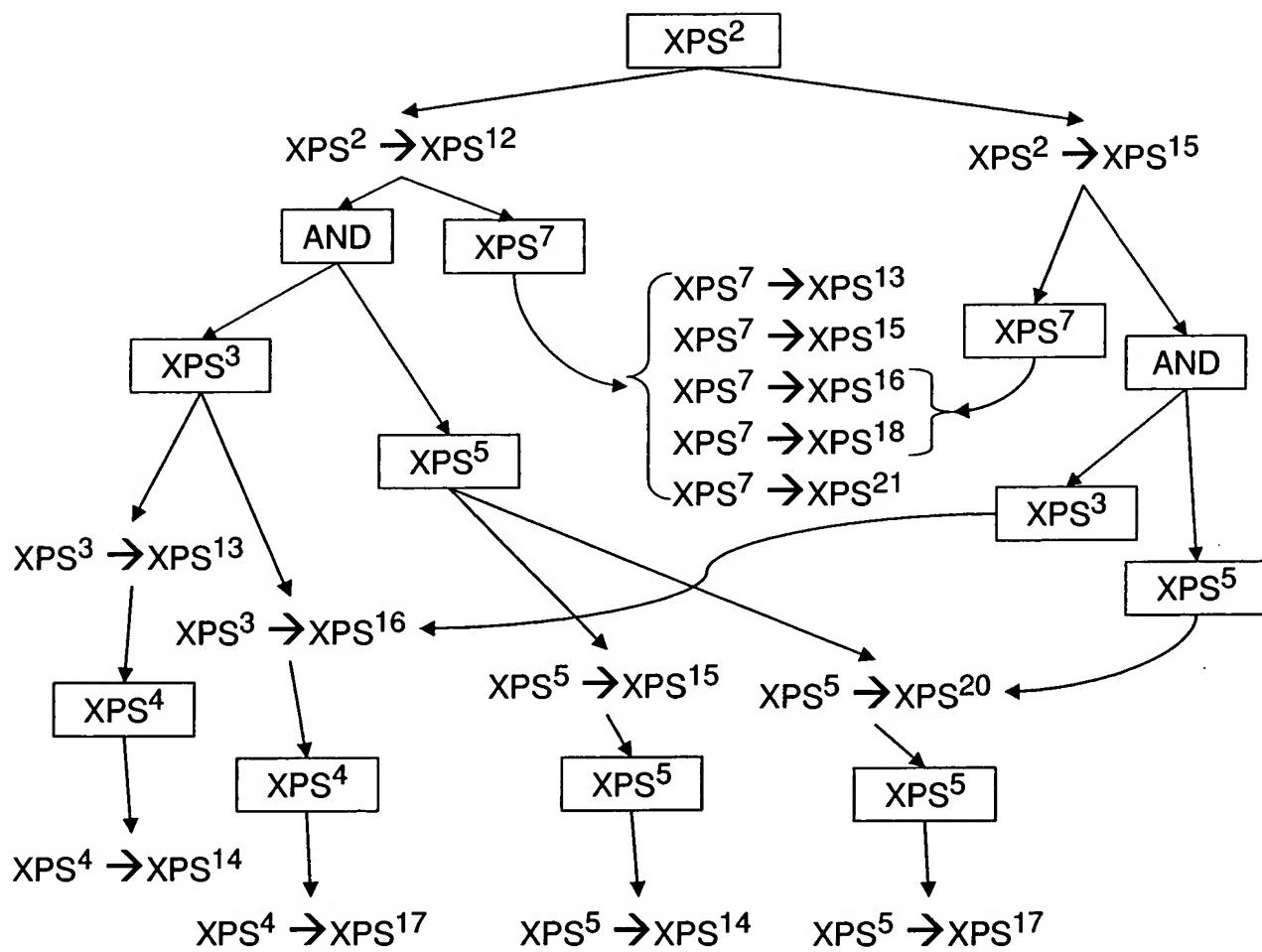


FIG. 9

REF/VALUE	PATH	TYPED VALUE
REF1	/A/B/C	5
REF2	/A/C	6
...

FIG. 10

FIG. 11

<i>Entry</i>	<i>PXR content</i>	<i>Extra Navigation</i>	<i>Predicates</i>	<i>Name/Anccestors</i>
1	(NR)	Forward/Reverse	As residual	As residual
2	(DR)	Full Query	Full Query	Full Query
3	(copy)	Forward	Below QEP, as residual	No
4	(NR, path)	Forward/Reverse	As residual	As pushdown
5	(DR, path)	Full Query	Full Query	As pushdown
6	(copy, path)	Forward	Below QEP, as residual	As pushdown
7	(NR, value)	Forward/Reverse	As pushdown	As pushdown
8	(DR, value)	Full Query	As pushdown	As residual
9	(copy, value)	Forward	As pushdown	Full Query
10	(NR, path, value)	Forward/Reverse	As pushdown	Below QEP, as residual
11	(DR, path, value)	Full Query	Full Query	As pushdown
12	(copy, path, value)	Forward	As pushdown	As pushdown

FIG. 12

